09/901,368   MS180587.01/MSFTP296US

REMARKS

Claims 1-19 are currently pending in the subject application and are presently under consideration. Claims 1 and 19 have been amended herein to further emphasize aspects of the subject invention. It is submitted that ADO.NET, the primary reference, does not constitute an enabling reference notwithstanding the proffered English translation, since the cited document does not enable one of ordinary skill in the art to produce applicants' claimed invention without undue experimentation. Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments made herein.

I.   Rejection of Claims 1-19 Under 35 U.S.C. §103(a)

Claims 1-19 stand rejected under 35 U.S.C. §103(a) as being unpatentable over ADO.NET (English Translation) and further in view of Omoigui US 2003/0126136 A1. This rejection should be withdrawn for at least the following reasons. ADO.NET and Omoigui, either alone or in combination, do not teach or suggest each and every aspect set forth in the subject claims.

> To reject claims in an application under §103, an examiner must establish a *prima facie* case of obviousness. A *prima facie* case of obviousness is established by a showing of three basic criteria. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. *Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations.* *See* MPEP §706.02(j). The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure. *See In re Vaeck,* 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

The claimed invention discloses a system and method for providing a streaming input and streaming output incremental XML transformer that can be employed in push

6

and/or pull model processing. The XML transformer provides for incrementally building output from XML data by loading a subset of an XML document into memory to perform a selective transformation. Independent claims 1 and 19 recite similar limitations, namely *a transformer that transforms one or more input XML items in a first format to one or more transformed XML items in one or more second formats, the one or more input XML items comprise a subset of XML items contained in a XML document.*

ADO.NET discloses an XMLReader and a component that provides XML, XSL, XSL/T, and X-Path tools that the XMLReader presumably utilizes to provide an XMLDataDocument. To the extent that it can be determined from the disclosure in the reference, the ADO.NET system receives XML documents as input, which allows for the composition of a XMLDataDocument object containing the individual XML documents. In the subject Office Action, it is incorrectly asserted that the reference discloses *transforming one or more input XML items in a first format to one or more transformed XML items in one or more second formats.* However, the reference does disclose such feature - the transformation illustrated is from XML data to a data set object expressed in a table structure (*See* page 4). As a result, the output of such operation is not an XML item, but rather a table entry within a data set. Although the reference discloses that the XML data within the XMLDataDocument object is checked for errors (*See* page 17), checking data within that object does not equate to *transforming* the data items from a first XML format to a second XML format as in applicants' claimed invention. IN accordance with the claimed invention, transforming XML items changes the actual representation of the XML items from one XML format to another XML format (*See e.g.,* page 10, lines 19-23), which is not disclosed or suggest by the reference. In view of the foregoing, it should be readily apparent that the cited reference does not teach or suggest *transforming one or more input XML items in a first format to one or more transformed XML items in one or more second formats.*

The Examiner further incorrectly asserts that checking XML documents within the XMLDataDocument object is equivalent to *a subset of XML items contained in a XML document*, as recited in the subject claims. As noted *supra,* the XMLDataDocument object is composed of individual XML documents, which were input to the ADO.NET system. The XMLDataDocument is created by reading the entire

contents of the XML data file (*See* page 16) and is subsequently synchronized with the DataSet contained in the system (*See* page 4), but the reference does not disclose reading or inputting a *subset of XML items contained in a XML document*. The Examiner incorrectly contended that checking the XML documents is synonymous with *the input XML items comprising a subset of XML items contained in a XML document*; however, checking of XML documents, as disclosed, is presumably nothing more than a verification step performed within the XMLDataDocument to ensure that each document has the proper syntax (*See* page 17). Furthermore, the reference does not explicitly disclose or imply that the checking of XML documents is limited to a *subset* of the XML documents; contrarily, because the system is checking XML documents for errors, all XML documents, rather than a *subset*, are presumably checked by the system. Accordingly, it is evident that ADO.NET does not teach or suggest *the one or more input XML items comprise a subset of XML items contained in a XML document*.

In addition, it is noted that in the Reply to Office Action dated February 28, 2005 and the Reply to Final Office Action dated August 2, 2005, it was stated that ADO.NET did not provide a comprehensible description of its activities sufficient to enable one of ordinary skill in the art to effectuate the disclosure therein without undue experimentation to produce the claimed invention. Applicants' representative maintains that despite the proffered English translation of ADO.NET the cited document does not provide an enabling reference for the purported teachings asserted by the Examiner.

Moreover, the Examiner concedes that ADO.NET does not teach or suggest *an output managing component that facilitates at least one of selectively pulling and pushing a subset of the transformed XML items prior to transforming all input XML items* and incorrectly relies upon Omoigui to make up for such deficiencies. Omoigui relates to an information retrieval and presentation framework for transferring information between client and server systems. In the section of the reference cited by the Examiner, Omoigui discloses a Query Manager and Results Browser that allow for information to be retrieved from a data store and displayed as a list of objects (*See* paragraph 0759). However, the Query Manager retrieves the information by querying the data store and receiving the results of that query as an entire set (*See* paragraph 0762). Thus, the query results are received as one complete set, rather than a *subset of items* received before the

8

complete request has been processed. Additionally, the Results Browser does transform the list of objects, however such a transformation outputs a new list of all the objects in the list, rather than *a subset of the transformed XML items prior to transforming all XML items*. Thus it is clear that Omoigui is silent with regard to *an output managing component that facilitates at least one of selectively pulling and pushing a subset of the transformed XML items prior to transforming all input XML items*.

In view of at least the foregoing, it is respectfully submitted that neither ADO.NET nor Omoigui, alone or in combination, teach or suggest each and every aspect of independent claims 1 and 19 (and claims 2-18 that depend there from). Therefore, it is respectfully requested that this rejection be withdrawn.
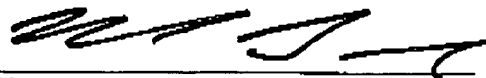
CONCLUSION

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063.

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number below.

Respectfully submitted,

AMIN & TUROCY, LLP

Himanshu S. Amin
Reg. No. 40,894

AMIN & TUROCY, LLP
24TH Floor, National City Center
1900 E. 9TH Street
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731